Confluence of Pure Differential Nets with Promotion

Paolo Tranquilli

Laboratoire PPS – Université Paris Diderot - Paris 7
Case 7014 – 75205 Paris – France

ptranqui@pps.jussieu.fr

Abstract. We study the confluence of Ehrhard and Regnier's differential nets with exponential promotion, in a pure setting. Confluence fails with promotion and codereliction in absence of associativity of (co)contractions. We thus introduce it as a necessary equivalence, together with other optional ones. We then prove that pure differential nets are Church-Rosser modulo such equivalences. This result generalizes to linear logic regular proof nets, where the same notion of equivalence was already studied in the literature, but only with respect to the problem of normalization in a typed setting. Our proof uses a result of finiteness of developments, which in this setting is given by strong normalization when blocking a suitable notion of "new" cuts.

1 Introduction

The inception of Linear Logic (LL, [1]) in the 80's has reinforced the bridge between logic and computer science already established by the Curry-Howard correspondence years before. LL is in fact a refinement of intuitionistic and classical logic brought forth by a fine semantical analysis. One of its main features is the introduction of two dual modalities, the *exponentials* ! and ?, regulating the use of structural rules (*weakening* and *contraction*), which on the program side correspond to erasure and duplication of resources.

This endeavour, among other things, led the way to a new, parallel syntax of proofs, *proof nets*. These are the syntax of choice for LL, especially when considering cut elimination. In fact one of the main advances of LL over classical logic is that, though preserving an involutive negation (and therefore two-sided sequents), it also preserves properties of intuitionistic logic lacking in the classical framework. One of these, central to our work, is confluence of cut elimination, i.e. the independence of the result of the cut elimination procedure with respect to the actual cuts one decides to reduce.

A further semantical analysis led by Ehrhard [2] has recently provided LL with new models based on topological vector spaces where we can take the derivative of an object. The efforts of the same author and Regnier have permitted to lift such operations to syntax, giving rise to Differential Linear Logic (DiLL, [3]), and their syntax, *differential nets*. Three new rules are introduced to handle the !-modality (*coweakening*, *cocontraction* and *codereliction*) which are duals to the LL rules handling ?. In the proofs as programs paradigm, codereliction allows

to introduce depletable resources, which may be asked for many times but may be used just one time, nondeterministically choosing which query they satisfy. This feature configures differential nets as a promising logical framework to extend the Curry-Howard correspondence to nondeterminism and concurrency (see [4]).

Actually, [3] gives the syntax for the promotion free fragment of DiLL only, giving rise to *differential interaction nets*, a nondeterministic example of Lafont's interaction nets [5]. By modelling nondeterminism by formal sums confluence remains an important property, which is however straightforward in an interaction net paradigm, where no reduction can change the other ones. Here we will extend such property to the whole of differential nets. Promotion in proof nets is handled by *boxes*, synchronized areas of proofs enabling to mark what is to be erased or duplicated. Boxes break the interaction net paradigm: there are cuts (the *commutative* ones) which can be changed by other reductions, so confluence is definitely more delicate.

Part of a previous work of ours [6] was focused on proving confluence for the intuitionistic fragment, which used the recursive types needed to translate λ-calculus. There we observed that confluence fails without keeping into account some semantically grounded equivalences, namely associativity of contractions and cocontractions. A fully quotienting syntax as the one used in [7] for LL is seemingly out of reach in DiLL. Our solution in [6] was employing generalized (co)contraction cells in the style of [8], and some additional reductions.

Here we generalize the result in three ways. By concentrating on the computational contents rather than the logical one, we consider *pure* nets, where types (i.e. formulae) play no role whatsoever, not even recursive ones. Furtherly, the needed equivalences are settled to the maximum extent by means of... equivalences on nets. We thus generalize the equivalences and reductions of [9], providing as a byproduct the first proof of confluence[1] for such LL proof nets with equivalences in the completely pure case, as previous works concentrated on normalization in the typed one. Finally, we are able to introduce one more equivalence potentially giving the right to always consider boxes without sums inside (the *bang sum* equivalence).

This result has several ramifications. As is evident in [10], this is the first step in proving strong normalization in the typed case[2]. Furtherly, as can be deducted from [9], this can be the ground for new work on calculi with explicit substitutions: whether by extending some results to untyped calculi; or by considering explicit substitutions for nondeterministic calculi akin to Boudol's λ-calculus with resources (see [6]).

Our technique, reminiscent of the work done on LL in [10], uses a finite development theorem used to prove a strong confluence property of a suitable notion of parallel reduction.

[1] To be precise, the stronger result of being Church-Rosser modulo (see Section 1.1).
[2] Actually the subject of a future submission by the author and Pagani.

1.1 Rewriting Theory Modulo Equivalence

The aim of this section is making the reader acquainted with the notion of rewriting modulo equivalence, to the extent needed for our purposes. We refer to [11, Section 14.3] for more indepth details and proofs.

Let $(S, \to)$ be an abstract reduction system and let $\sim$ be an equivalence relation on S. As usual, $\xrightarrow{=}$ and $\xrightarrow{*}$ denote the reflexive and reflexive-transitive closures of $\to$ respectively. Take also a symmetric relation $\vdash$ such that $\vdash^* = \sim$, possibly $\sim$ itself. Let $s \between t$ (t and s are **joinable modulo** $\sim$) if $s \xrightarrow{*} \sim \xleftarrow{*} t$. We say then that $\to$ is

- locally confluent modulo $\sim$ if $\leftarrow \to \subseteq \between$;
- confluent modulo $\sim$ if $\xleftarrow{*} \sim \xrightarrow{*} \subseteq \between$;
- locally coherent with $\vdash$ if $\vdash \to \subseteq \between$;
- Church-Rosser modulo $\sim$ (or $\mathsf{CR}\sim$) if $\approx \subseteq \between$, where $\approx := (\to \cup \leftarrow \cup \sim)^*$;
- strongly normalizing modulo $\sim$ (or $\mathsf{SN}\sim$) if $\rightsquigarrow$ is SN, where[3] $\rightsquigarrow := \sim\to\sim$;
- strongly Church-Rosser modulo $\sim$ if $\sim \xleftarrow{=} \xrightarrow{=} \sim \subseteq \xrightarrow{=} \sim \xleftarrow{=}$;

The last definition is our terminology, while the rest follows [11]. Being Church-Rosser modulo $\sim$ is the most important property of all those concerning confluence. In particular it implies the unique normal form modulo $\sim$ property, ($\approx = \sim$ on normal forms), which again implies that in order to compute the normal form one can use just regular reductions, without ever be forced to $\sim$-convert in order to get the result[4]. Contrary to what happens in regular reduction, $\mathsf{CR}\sim$ is strictly stronger than plain confluence in absence of WN [11, Remarks 14.3.6, Exercise 14.3.7]. Following are some important lemmas: the first is a generalization of Newman's Lemma, the last is a trivial result we did not find in the literature which we will need in our proof.

Lemma 1 (Huet). *If $\to$ is $\mathsf{SN}\sim$, locally confluent modulo $\sim$ and locally coherent with $\vdash$, then it is $\mathsf{CR}\sim$.*

Lemma 2 (van Oostrom). *$\to$ is $\mathsf{CR}\sim$ iff $\xrightarrow{*}$ is strongly $\mathsf{CR}\sim$.*

Lemma 3. *If $\to$ is strongly $\mathsf{CR}\sim$, then it is $\mathsf{CR}\sim$.*

Proof. Straightforward induction: to show that $\sim \xleftarrow{*} \xrightarrow{*} \sim$ is joinable, we proceed by induction first on one side, then on the other.

2 The System

A **net** is intuitively a network of **cells** linked by **wires** connecting their **ports**. A little more formally, a net π is given by the following data.

[3] Here like in the rest of the paper, := means "defined as".

[4] This also mean there is never the need to perform conversion steps in order to *ready* some cuts, i.e. make them visible.

- A set $\mathsf{p}(\pi)$ of ports.
- A set $\mathsf{c}(\pi)$ of cells; to each cell c is assigned a **symbol** $\sigma(c)$ in a given alphabet, a port in $\mathsf{p}(\pi)$ called **principal**, and a number of other, **auxiliary** ones. How the latter are treated distinguishes between two kinds of cells: in non commutative ones, auxiliary ports are a finite sequence, in **commutative** ones they form a finite set. Every port in $\mathsf{p}(\pi)$ can be associated with at most one cell; a port associated with a cell is called **connected**, otherwise it is **free**. Free ports (also called conclusions) are denoted by $\mathsf{fp}(\pi)$. The number of auxiliary ports is determined by the symbol $\sigma(c)$.
- A set $\mathsf{w}(\pi)$ of wires, which can be either unordered pairs $\{p, q\}$ of ports, or **deadlocks**, i.e. wires not connecting any port (intuitively short circuited wires). Each port is in exactly one wire. A **directed wire** is an ordered pair (p, q) such that $\{p, q\}$ is a wire. **Terminal wires** are the directed ones going to the free ports.

An **elementary path** in π is one in the graph trivially obtained by taking cells and free ports as nodes and directed wires as edges, which moreover does not intersect itself[5]. A **polynet** is a formal sum of nets, or equivalently a multiset of nets, all sharing the same free ports. At times we distinguish nets (thus singletons) from polynets by calling them **simple**.

2.1 Statics

DiLL_0 nets and polynets are built from all the symbols in Figure 1 but the box one. These are exactly the differential interaction nets presented in [3]. For the moment let us ignore the labels we assign to the ports in the figure, which will be needed only later (see page 9). As usual, the apex of the cell represents the principal port, while the auxiliary ones are depicted on the opposite side.

In order to add boxes, one proceeds by induction, by considering them as cells having a whole polynet as symbol. Let DiLL_{k+1} nets and polynets be the ones built from all the cells of Figure 1 where for each box its symbol is a polynet π in DiLL_k and there is a bijection between its ports and $\mathsf{fp}(\pi)$. The symbol $\sigma(B)$ of a box B is also called its **contents**. We will denote by $!\pi$ a generic box having π as contents. A DiLL polynet π is one of DiLL_k for any k; if such k is minimal, we say that k is the depth of π (in fact, the maximal number of nested boxes). A port is **active** if it is either a principal one, or an auxiliary one of a box. A wire linking two active ports is a **cut**.

Figures 4 and 5 will show examples of differential nets. The explicit marking of ports is dropped as they can always be identified with the extremities of wires.

Let $\mathsf{p}_!(\pi)$, $\mathsf{fp}_!(\pi)$ $\mathsf{c}_!(\pi)$ and $\mathsf{w}_!(\pi)$ be the set of all occurrences of ports, free ports, cells and wires respectively occurring in π, including in all the contents of the boxes in π. We can slice those sets by depth, so we will denote by $\mathsf{p}_i(\pi)$,

[5] Technically, one prohibits the repetition of unoriented wires and that three ports of the same cell be crossed by the path.

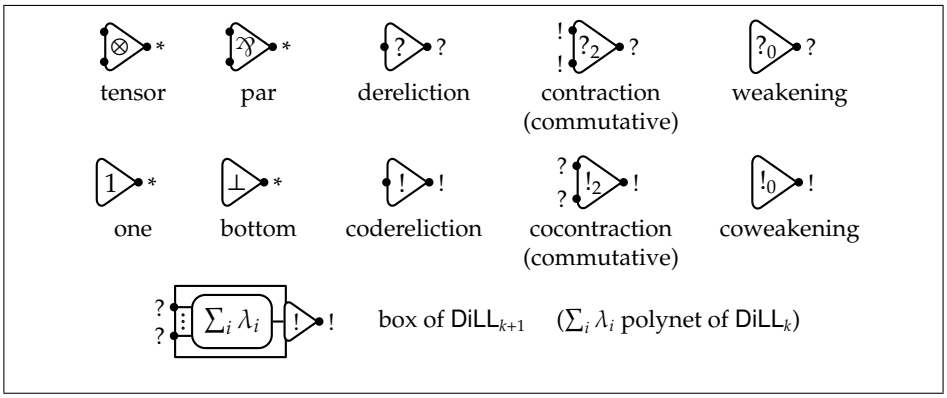


Fig. 1: The cells of differential nets. The labels in $\{!, ?, *\}$ assigned to ports will be needed only later (see page 9).

$\mathsf{fp}_i(\pi)$ $\mathsf{c}_i(\pi)$ and $\mathsf{w}_i(\pi)$ the corresponding elements of the nets contained in i nested boxes, where i is called the depth of the element in π[6].

Correctness criterion. As usual, the nets blindly built with the cells available are not in general "correct", where the word can take the meaning of unsequentializable in sequent calculus, or having deranged computational behaviour. Since [12] one of the most used correctness criteria for proof nets is that of switching acyclicity. Given a DiLL net, a **switching path** is an elementary one which does not traverse two auxiliary ports of any $\parr$ or contraction cell (does not bounce "above" it). A DiLL polynet is called a DiLL **proof net** (or differential proof net) if it is switching acyclic, i.e. it has no deadlocks nor switching cycles, and inductively all box contents are also switching acyclic. In particular from now on we can suppose all our polynets to be deadlock-free.

2.2 Dynamics

As with various calculi, the reduction of differential nets can be defined as the context closure of a set of reduction rules, presented as pairs of redexes and contracta. A **linear context** $\delta[\ ]$ is a simple net δ together with a subset H_δ of its free ports (the **hole** of $\delta[\ ]$). It is linear as it is not a sum and the hole is not inside a box. Given a simple net λ and a bijection σ between H_δ and $\mathsf{fp}(\lambda)$, the plugging $\delta[\lambda]$ of a simple net λ in the hole of $\delta[\ ]$ amounts to identifying the

[6] All of this can be defined more formally by an inductive definition. Nevertheless we leave it to the reader as an easy exercise.

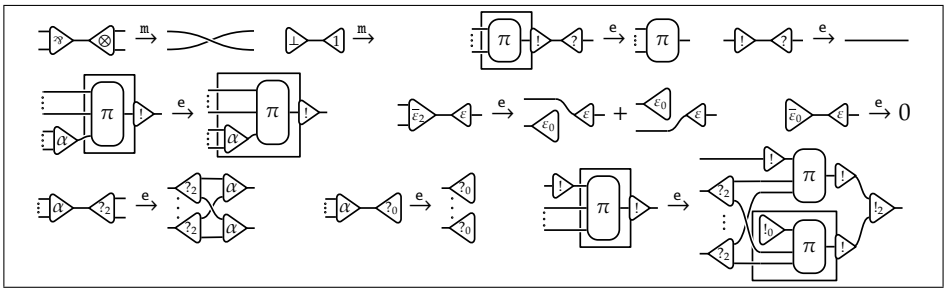
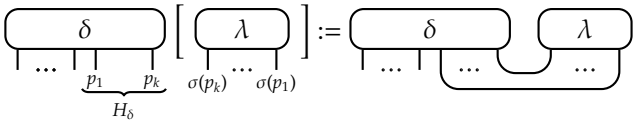


Fig. 2: The multiplicative and exponential reduction rules of DiLL. ε and $\overline{\varepsilon}$ denote either ? and ! or vice versa. α denotes any symbol among $!_2$, $!_0$ or a box symbol π. In particular weakening against coweakening reduces to the empty net. We make implicit use of the rule for context plugging of sums: the π inside boxes is a polynet. For example the dereliction on box rule may introduce sums.

ports according to σ and welding the wires that come together in this way[7].



This definition is then extended by linearity when we plug a polynet, by setting $\delta\left[\sum_i \lambda_i\right] := \sum_i \delta[\lambda_i]$.

Finally, **contexts** generalize the concept in the following way. A linear context $\delta[\ ]$ is a context; furthermore if $\omega[\ ]$ is a context then:

- $\delta[\omega[\ ]]$ for $\delta[\ ]$ linear is a context[8];
- $\omega[\ ] + \pi$ for π polynet is a context;
- $!\omega[\ ]$ (i.e. a box containing a context) is a context.

Plugging is easily extended to all contexts. The **context closure** $\tilde{R}$ of a relation R is then defined by $\pi\ \tilde{R}\ \sigma$ iff $\pi = \omega[\lambda]$, $\sigma = \omega[\mu]$ and $\lambda\ R\ \mu$.

We are now able to define multiplicative reduction $\xrightarrow{\mathtt{m}}$ and the exponential one $\xrightarrow{\mathtt{e}}$ by context closure of the rules of Figure 2, which are pairs consisting of a simple net (the **redex**) and a polynet (the **contractum**). Each redex here is identified by a unique cut. The union $\xrightarrow{\mathtt{me}}$ of the two reduction is the cut elimination of DiLL.

2.3 Equivalences and Canonical Reductions

As we will show with Remark 5, the reductions just presented fail to give a confluent system: we cannot ignore associativity of (co)contractions and neutrality

[7] For the quite delicate technical details the reader is referred to [13].

[8] Composition of contexts should be defined, but it is trivial once plain plugging is defined.

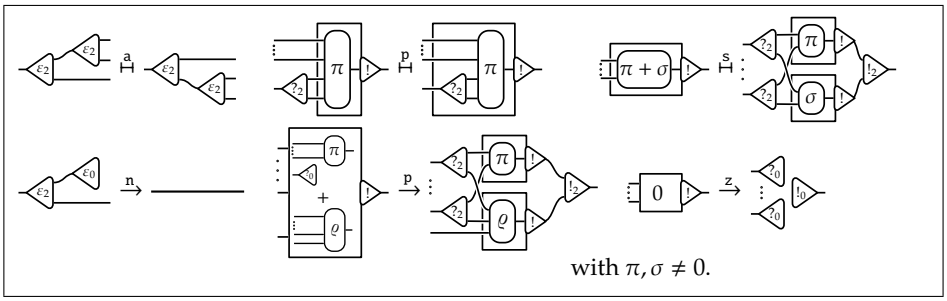


Fig. 3: Top: the rules for associative equivalence $\overset{\mathtt{a}}{\sim}$, the push one $\overset{\mathtt{p}}{\sim}$ and the bang sum one $\overset{\mathtt{s}}{\sim}$; $\mapsto$ denotes a one-step conversion. Bottom: the rules for neutral reduction $\overset{\mathtt{n}}{\to}$, the pull one $\overset{\mathtt{p}}{\to}$ and the bang zero one $\overset{\mathtt{z}}{\to}$. The condition $\pi, \sigma \neq 0$ applies to all rules.

of (co)weakening over (co)contraction. This prompts us to introduce the former as an equivalence and the latter as a reduction. As we need anyway to consider reduction modulo an equivalence, we also study other equivalences (backed by semantical and observational equivalence) which are optional though must be taken together. Each equivalence is accompanied by a reduction which in a sense settles a zeroary case of the equivalence. Reversing each of these gives unwanted looping reductions. The **associative**, **push** and **bang sum** equivalences, together with the **neutral**, **pull** and **bang zero** reductions (which *do not* reduce cuts), are shown in Figure 3. The $\pi, \sigma \neq 0$ condition is needed: without one would be able to spawn arbitrarily large trees of contractions, easily giving again unwanted looping reductions. From now on $\sim$ and $\overset{\mathtt{c}}{\to}$ (**canonical** reduction) will denote either $\overset{\mathtt{a}}{\sim}$ and $\overset{\mathtt{n}}{\to}$ or the union of the $\mathtt{a}$, $\mathtt{p}$ and $\mathtt{s}$ conversions and the $\mathtt{npz}$-reduction respectively.

Proposition 4 (stability of correctness). *If π is switching acyclic and $\pi \overset{\mathtt{mec}}{\longrightarrow} \pi'$ or $\pi \sim \pi'$ then π' is switching acyclic also.*

Proof. Many cases belong to the literature of LL and to [3]. The new ones are straightforward checks.

Let us spend some more words on $\sim$ and $\overset{\mathtt{c}}{\to}$. The push equivalence[9] has already been studied in the literature on proof nets and explicit substitutions [8,9]. The pull reduction may seem somewhat complicated, however it is a generalization of the reduction pulling out weakenings from boxes [9]. The usual reduction can be reobtained when having $\varrho = 0$, which by means of a $\mathtt{z}$-reduction and some $\mathtt{n}$ ones gives the expected result. Such form (which in fact contains a sort of on-the-fly $\mathtt{s}$-conversion) is required in order to get local coherence.

[9] The name may be misleading, as an equivalence is not directed. It comes from [14] and [6] where it was treated as a reduction, and we felt like keeping it because of the good name pairing with the pull reduction.

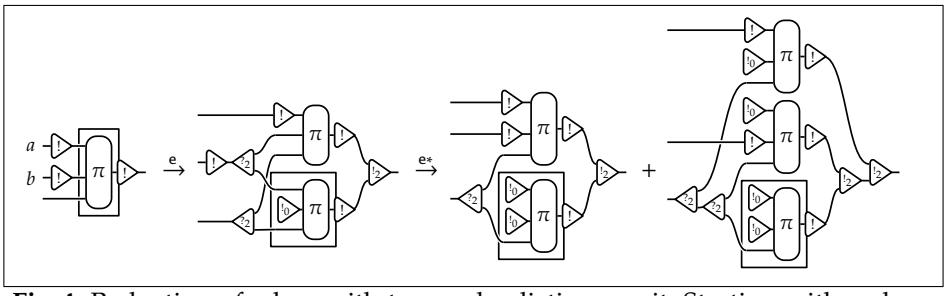
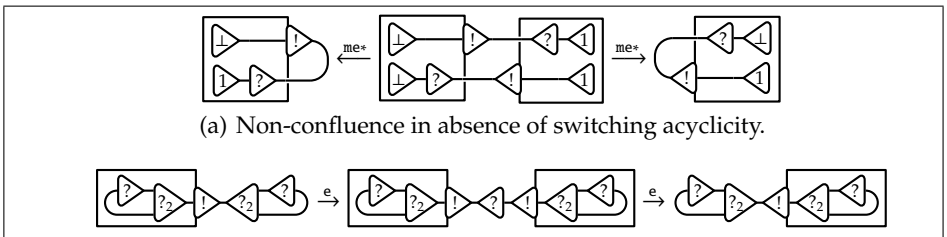


Fig. 4: Reduction of a box with two coderelictions on it. Starting with coderelliction b swaps the two linear copies of $!\pi$ and therefore both the cocontraction and contraction trees in the last addend.



(a) Non-confluence in absence of switching acyclicity.

(b) Non-termination of exponential reduction with switching acyclicity.

Fig. 5: Issues with confluence. Figure 5(a) shows the need for correctness. The example shown is even simply typed. Figure 5(b) shows how in the pure case even the exponential reduction alone is not terminating.

The part about sums inside boxes was already known to be valid semantically and observationally: we give here some sintactic ground to using it. From the point of view of semantics it is interesting to note that it implements the well known exponential isomorphism $!A \otimes !B \cong !(A \,\&\, B)$ from linear logic (see [2]).

Examples and remarks. Figure 4 gives the reason to employ associative equivalence, showing the reduction of the coderelictions on box critical peak. One reduction only is shown, as the other is symmetric. One can already see two big differences with respect to LL and the work done with it in [10]: firstly, sums may arise even without the "logical" step of dereliction on box; moreover, the coderelliction on box rule, which reduces a commutative cut, changes the possible cuts on *all* other cuts of the box. These problems prevent an immediate adaptation of the measures used in [10]. The nets in Figure 5 are examples already known in LL showing issues about correctness and types.

Remark 5. The net shown in Figure 4 is a counterexample to DiLL being confluent without $\overset{a}{\sim}$. Other similar diagrams show we cannot also omit the n reduction.

3 The Finite Development Theorem

In [15], Danos proved the counterpart of the finite development theorem for MELL, and Pagani and Tortora de Falco did the same for the whole of second order LL in [10]. In this setting the actual definition of what a "new" redex is gets more technical.

3.1 Marking New Cuts

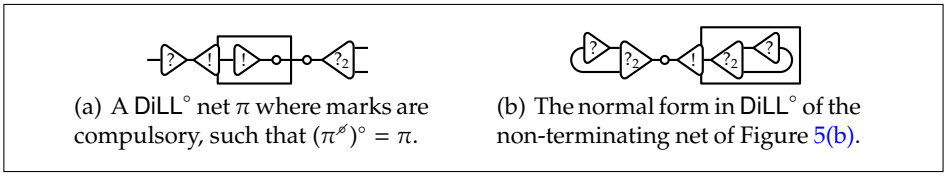
We define a notion of new and old cuts, by leaving a mark on the new ones. **Marks** are cells of a new symbol with two ports and no reduction rule, graphically depicted by little circles. Its main purpose is to block reductions and equivalences (for example a mark between two contractions blocks the a-conversion).

Ideally, these marks are placed during reduction to block "new" wires. By new we mean two kinds of wires: those that in a typed setting would decrease the logical complexity of the cut formula, and those that before the reduction were exponential clashes. The latter are peculiar to a truly untyped setting, and are brought by the opening box and neutral reductions, which erase an exponential port. For example, if we erase marks from the net shown in Figure 6, and we fire the dereliction against box redex we end up with a valid multiplicative cut which was a clash before. Rather than lock this special kind of "new" wires during reduction, we can lock all potentially dangerous clashes since the start, as markings will prevent new clashes from arising. We thus need to define what an exponential clash is.

Let τ be the partial function from $\mathsf{p}_!(\pi)$ to the labels $\{!, ?, *\}$ thus defined. On ports of cells it gives the values already shown in Figure 1; on the ports of marks it is undefined; for $p \in \mathsf{fp}_!(\pi)$ we set $\tau(p) = ?$ if p is over an auxiliary port of a box, we leave it undefined otherwise. τ provides for a sort of pre-typing. A directed wire (p, q) is called a **!-wire** (resp. **?-wire**) if $\tau(p) = ?$ and $\tau(q) = !$ (resp. vice versa), where however we let at most one of the two be undefined. In any case ! and ?-wires are called **exponential** (which applies to undirected wires also). An **exponential clash** (simply clash from now on) is a wire $\{p, q\}$ such that one of $\tau(p)$, $\tau(q)$ is ! (resp. ?) and the other is defined but not ? (resp. !)[10].

DiLL$^\circ$ is the system given by polynets with marks and without clashes, and by changing some rules to introduce the mark as depicted in Figure 7. It is immediate to see that the absence of clashes is preserved by reduction, as the new wires which could bring close unmatched ports are interrupted by marks. From the point of view of DiLL, clash-freeness imposes just that some marks be added: given any DiLL polynet π, we define the injection π° in DiLL$^\circ$ by placing a mark interrupting each clash. Conversely, DiLL$^\circ$ can be clearly surjected on DiLL by erasing all marks. We call this surjection $\pi^\not\circ$. The net π in Figure 6(a) is an example of DiLL$^\circ$ net enjoying $(\pi^\not\circ)^\circ = \pi$. On the other hand, if σ is the net in Figure 5(b), then $\sigma^\circ = \sigma$ and it is strongly normalizing to the net in Figure 6(b).

[10] More intuitively: !-wires and ?-wires are those that in a typing attempt would get an outermost ! or ? respectively, while clashes would give a failure to unify an outermost exponential modality.



(a) A DiLL$^\circ$ net π where marks are compulsory, such that $(\pi^{\not\circ})^\circ = \pi$.

(b) The normal form in DiLL$^\circ$ of the non-terminating net of Figure 5(b).

Fig. 6: Examples for DiLL$^\circ$.

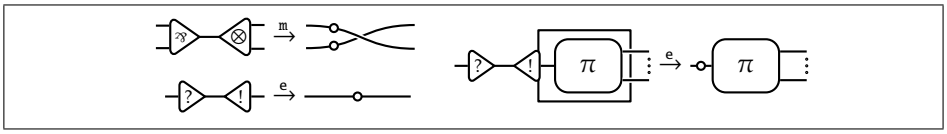


Fig. 7: The modified reduction rules of DiLL$^\circ$.

3.2 Measuring Exponential Reduction

Ideally, we may regard exponential reduction as a procedure that "slides" cells along exponential paths in the net, with ! and ? cells sliding in opposite direction. We thus assign to each cut a natural number, indicating how far the two cells around it are from the end of the path they are sliding on. After a reduction however many cuts may have arisen. So we will employ the multiset of the weights of the cuts and the multiset order[11]. Global additive duplication poses another problem. In [6] we settled it by employing multisets of multisets. Here however we estimate how many addends can sprout during reduction, so we can use this value and count each cut as many times as there can be addends containing it. We will also need to get an estimate of the number of copies (both regular and linear) of a box.

Exponential paths. An **!-path** (resp. **?-path**) is an elementary path made only of !-wires (resp. ?-wires), not traversing any mark, dereliction or codereliction (though it may end on them). In either case, the path is called **exponential**. All cells internal to an exponential path must necessarily be contractions, coco-ntractions or boxes. The main technical advantage of DiLL$^\circ$ over DiLL is that in it no reduction can open new exponential paths.

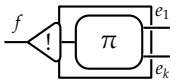
Next we define by mutual induction three basic measures on which we will base the measure of the whole net. Two of them, the **?-weight** $?(e)$ and the !-weight $!(e)$, are on wires. The third, the **spread** $\mathrm{sp}(\lambda)$, is defined on simple nets.

Weighting wires and estimating addends. First, $?(e) = !(e) = 1$ if e is not exponential; otherwise, let e be oriented so that it is an !-wire (resp. ?-wire), and let c be the node (either a free port or a cell) to which e thus points. Table 1 provides the laws for $?(e)$ (resp. $!(e)$), giving them depending on c. By e^λ we denote the wire

[11] Multisets over a well founded set, presented as $[a_1, \ldots, a_k]$ and additive notation, have the well founded order given by the transitive closure of $A + [a] > A + B$ with $\forall b \in B : b < a$.


: $?(e) = ?(e_1) + ?(e_2)$;


: $?(e) = ?(f)$;

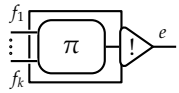

: $?(e_i) = \begin{cases} ?(f)(1 + \sum_j !(e_j)) & \text{if } \pi = 0, \\ ?(f)(1 + \sum_j !(e_j)) \sum_{\lambda \in \pi} \mathrm{sp}(\lambda) ?(e^\lambda) & \text{otherwise;} \end{cases}$

otherwise: $?(e) = 1$.

: $!(e) = !(f)$ if $p \in \mathsf{fp}_0(\sigma(B))$ for a box B, p is above an auxiliary port, and f is the wire corresponding to p outside the box,


: $!(e) = !(e_1) + !(e_2)$;


: $!(e) = !(f)$;


: $!(e) = \begin{cases} 1 + \sum_j !(f_j) & \text{if } \pi = 0, \\ (1 + \sum_i !(f_i)) \sum_{\lambda \in \pi} \mathrm{sp}(\lambda) & \text{otherwise.} \end{cases}$

otherwise: $!(e) = 1$.

$$\mathrm{sp}(\lambda) = \prod_{\substack{c \in \mathsf{c}_0(\lambda) \\ \sigma(c) = ?}} !(c) \cdot \prod_{\substack{c \in \mathsf{c}_0(\lambda) \\ \sigma(c) = !}} ?(c)$$

Table 1. Rules for the ?-weight (top), the ! one (middle) and the spread sp (bottom). We remind that the subscripts of the products for the spread make them range over the derelictions and coderelictions at depth 0.

corresponding to e inside a box, in the net λ of the box contents. At the bottom we also show the law for the spread. Notice that there is a circular dependency between the three measures, so the next lemma is not trivial.

Lemma 6. *Given a* DiLL$^\circ$ *proof net* π, $?(e)$, $!(e)$ *and* $\mathrm{sp}(\lambda)$ *are defined and unique for all* $e \in \mathsf{w}_!(\pi)$ *and all* λ *simple subnets of* π *at any depth.* →*tech.app.*

Proof (sketch). Substituting the free port cases by assigning variables, we get all the measures as polynomials in those variables (see the section on modularity). One then proceeds by a primary induction on the depth: supposing all the (polynomial) measures have been defined on strictly less depth, one can

– define $!(e)$ by induction on the maximal length of ?-paths starting from e (instantiating the variables of the ?-conclusions inside boxes in the process);

– only then, define $?(e)$ by induction on the maximal length of !-paths (relying also on measures at greater depth just instantiated);
– finally define the spread from the two. □

Weighting nets and polynets. The **weight** $|e|$ of a wire is $?(e) + !(e)$. Let $!\mathsf{cw}_0(\lambda)$ (resp. $\mathsf{b}_0(\lambda)$) be the set of exponential cuts (resp. boxes) at depth 0 of a simple net λ. Let us fix a polynet π. Then for each sum (i.e. multiset) of subnets of π we define by induction on their depth the following multiset (λ will denote a generic simple subnet):

$$\|\sigma\| := \sum_{\lambda\in\sigma} \mathrm{sp}(\lambda)\,\|\lambda\|\,, \quad \|\lambda\| := [\,|e| \mid e \in !\mathsf{cw}_0(\lambda)\,] + \sum_{B\in\mathsf{b}_0(\lambda)} \#(B)\,\|\sigma(B)\|\,,$$

where $\#(B)$, the **count** of the box, is $?(e)(1 + \sum_j !(f_j))$ with e and f_j the wires on the principal and the auxiliary ports respectively. In particular we just defined the measure for the whole net.

Notice that this measure depends monotonously from the weight of each part of the net. This intuition will be given a solid ground by the modularity Lemma 8.

Intuitive ideas of the measures. Morally $!(e)$ measures the size of the tree of cocontractions above e (which is invariant under associativity). The most important feature is that it counts all the coderelictions linked to e. On boxes we count

– the !-weight on the auxiliary ports because the codereliction against box rule creates a contraction and a codereliction; plus one to count the box itself, especially if it has no auxiliary ports;
– multiplied by the spread of the contents in order to be invariant by s-conversion, and keep such invariants even if the sum inside. . . spreads.

Dually $?(e)$ measures the size of the tree of contractions above e. The rule when e is on an auxiliary port of a box B contains:

– $?(f)$ because the contractions on the principal port of B may shift to auxiliary ports during reduction;
– the sum of !-weights of the auxiliary wires because codereliction against box creates contractions; plus 1 to provide something to decrease when a cut enters a box (box against box and those similar);
– the ?-measures inside because either by opening the box or by p-conversion the contraction trees inside can pour outside; summed to respect both p-conversion and s-conversion; however the sum is weighted with the internal spread to avoid that a reduction generating a sum inside could raise such weight.

As already hinted, $\mathrm{sp}(\lambda)$ estimates how many addends may have a reduct of λ. This is achieved by morally multiplying all the possible number of choices potentially to be done in λ. Now sums arise

– on (co)dereliction against co(co)ntraction reductions, so the size of the tree of co(co)ntractions on the principal port of a (co)dereliction should estimate what choices that (co)dereliction may do;
– on (co)dereliction against box rules, when the box contains an actual sum; however the spread of a box contents are already accounted for in both "moral" contraction and cocontraction trees.

Finally the cuts inside a box B count $\#(B)$ times as this number estimates how many regular and linear copies of its contents may be done, and all cuts count $\mathrm{sp}(\lambda)$ times to account for additive duplication.

Replacement and modularity lemmas. For the purpose of working modularly with the measure, we introduce variables on the terminal wires over $\mathsf{fp}_0(\pi)$. So for each terminal ?-wire d (resp. !-wire), we consider *variables* $!(d)$ (resp. $?(d)$). All measures become then polynomials with natural coefficients; in the following, we will consider the extensional (i.e. pointwise) order $\leq$ on non-zero values for such variables.

For different simple nets λ, μ, we distinguish the weights calculated on one or the other by putting them as superscripts, as in $?^{\lambda}(e)$. Suppose λ and μ are two nets with identified terminal wires C. We say that λ **can replace** μ if for each terminal wire d we have that $!^{\lambda}(d) \leq !^{\mu}(d)$ and $?^{\lambda}(d) \leq ?^{\mu}(d)$ (one of the comparisons may be a trivial one among the same variables).

Lemma 7 (replacement). *Suppose λ can replace μ, $\omega[\ ]$ is a linear context with $\omega[\lambda]$ and $\omega[\mu]$ proof nets. Then for each e wire in the context ω, $?^{\omega[\mu]}(e) \leq ?^{\omega[\lambda]}(e)$ and $!^{\omega[\mu]}(e) \leq !^{\omega[\lambda]}(e)$.* →*tech.app.*

Proof (sketch). By easy induction on $\omega[\ ]$. One gradually moves cells from ω to λ and μ, and then all the wires which loop on the hole of ω (where the acyclicity condition saves us from circular dependencies).

In the following the weight $|D|$ of a set of wires D is the multiset of the weights of its wires. A terminal wire is **dormant** if it connects an active port or two free ones. Dormant wires are those that can become cuts when glued in a context.

Lemma 8 (modularity). *Take λ and $\mu_1, \ldots, \mu_n$ with simple modules and $\omega[\ ]$ a context such that $\omega[\lambda]$ and $\omega[\mu_i]$ are all* DiLL$^\circ$ *pure proof nets. Suppose the following points are satisfied:*

– *for every i we have that μ_i can replace λ;*
– $\sum_i \mathrm{sp}(\mu_i) \leq \mathrm{sp}(\lambda)$*;*
– *if $n = 0$, then $\|\lambda\| > [\,]$; otherwise, for every i we have $\left\|\mu_i\right\| + \|D_i\|^{\mu_i} < \|\lambda\|$, (resp. $\leq$) where D_i is the set of active wires in μ_i that are not dormant in λ.*

Then we have the pointwise inequality $\left\|\omega\left[\sum_i \mu_i\right]\right\| < \|\omega[\lambda]\|$ (resp. $\leq$). →*tech.app.*

Proof (sketch). By induction on the depth of the hole in $\omega[\ ]$. The base step is little more than the replacement lemma. The inductive step shows that the deepest box where the substitution happens satisfies all the hypotheses.

Lemma 9. $\|\,.\,\|$ *has the following properties.*

- *if* $\pi \xrightarrow{\mathtt{e}} \pi'$ *then* $\|\pi'\| < \|\pi\|$;
- *if* $\pi \xrightarrow{\mathtt{mc}} \pi'$ *then* $\|\pi'\| \leq \|\pi\|$;
- *if* $\pi \sim \pi'$ *then* $\|\pi'\| = \|\pi\|$. →*tech.app.*

Proof (sketch). Thanks to the modularity lemma, it suffices to consider each pair of redex and contractum, assign variables to their terminal wires as explained, and mechanically check the hypotheses of the lemma. For equivalences it is enough to check the same hypotheses replacing inequalities by equalities.

Theorem 10 (finite developments). *Reduction on* DiLL$^\circ$ *is* SN. →*tech.app.*

Proof (sketch). Only $\mathtt{m}$ and $\mathtt{c}$ remain to be settled. For all reductions, the pair given by $\left(\|\pi\|, \#_{\mathtt{m}}(\pi) + \#_{\mathtt{c}}(\pi)\right)$ strictly decreases for lexicographic ordering, where $\#_{\mathtt{m}}$ just counts the multiplicative cells in π, and $\#_{\mathtt{c}}$ weights coweakenings, weakenings and boxes contining 0 in the following inductive way:

$$\#_{\mathtt{c}}(\pi) := 1 + \#_{!_0}(\pi) + \#_{?_0}(\pi) + \sum_{B \in \mathsf{b}_0(\pi)} (1 + \deg(B))\, \#_{\mathtt{c}}(\sigma(B))$$

where $\#_{!_0}$ and $\#_{?_0}$ count coweakenings and weakenings at depth 0, and the degree $\deg(B)$ is the number of ports of B. This accounts both for 0-boxes becoming (co)weakenings, and for weakenings pulls splitting boxes. $\#_{\mathtt{c}}$ by the way can be used to prove that $\mathtt{c}$-reduction is strongly normalizing.

4 Proving Confluence

Recall that $\sim$ and $\mathtt{c}$ may be $\mathtt{a}$-equivalence and $\mathtt{n}$-reduction, or full $\mathtt{asp}$-equivalence and $\mathtt{nzp}$-reduction. Some of the diagrams show we cannot separate $\mathtt{s}$-equivalence from the $\mathtt{p}$ one (and their associated reductions).

Proposition 11. *Reduction on* DiLL$^\circ$ *is* CR$\sim$. →*tech.app.*

Proof (sketch). The proof relies on Lemma 1 and the strong normalization of DiLL$^\circ$ we proved in the previous section. One has therefore to prove local confluence modulo $\sim$ and local coherence with the generating relation $\mapsto$, by going through all the (numerous) critical pairs. As for multiplicative reductions, we can even be more specific, as $\mathtt{m}$ is strongly confluent and commutes with $\mathtt{e}$ (in particular $\xleftarrow{\mathtt{m}}\xrightarrow{\mathtt{e}} \subseteq \xrightarrow{\mathtt{e}}\xleftarrow{\mathtt{m}^*}$).

Figure 4 has already shown one of the most interesting cases for local confluence, two coderelictions on a box. The other critical pairs for both confluence and coherence can be joined by a long exercise in reductions.

We can finally arrive to the main theorem of the work.

Main Theorem. *Reduction on* DiLL *pure proof nets is* CR$\sim$*, and so are* $\mathtt{m}$ *and* $\mathtt{ec}$ *taken alone.*

Proof. Using DiLL$^\circ$ we define a parallel reduction $\twoheadrightarrow$. Let $\pi \twoheadrightarrow \sigma$ iff $\pi^\circ \xrightarrow{\texttt{mec}*} \varrho$ in DiLL$^\circ$ and $\sigma = \varrho^{\not\circ}$. Then

- $\xrightarrow{\texttt{mec}} \subseteq \twoheadrightarrow \subseteq \xrightarrow{\texttt{mec}*}$, so that $\twoheadrightarrow^* = \xrightarrow{\texttt{mec}*}$ (notice π° cannot block any reduction);
- $\twoheadrightarrow$ is strongly CR$\sim$ because $\xrightarrow{\texttt{mec}*}$ is so in DiLL$^\circ$.

Then we conclude, as $\twoheadrightarrow$ is CR$\sim$ by Lemma 3 ($\twoheadrightarrow$ is reflexive), which means that $\twoheadrightarrow^*{=}\xrightarrow{\texttt{mec}*}$ is strongly CR$\sim$, which in turn by Lemma 2 gives Church-Rosser modulo $\sim$ for the ordinary reduction[12]. It is not hard to give parallel reductions for the ec and m ones and do the same.

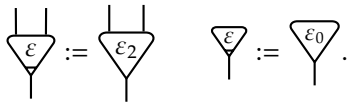
References

1. Girard, J.Y.: Linear logic. Th. Comp. Sc. **50** (1987) 1–102
2. Ehrhard, T.: Finiteness spaces. Mathematical. Structures in Comp. Sci. **15**(4) (2005) 615–646
3. Ehrhard, T., Regnier, L.: Differential interaction nets. Theor. Comput. Sci. **364**(2) (2006) 166–195
4. Ehrhard, T., Laurent, O.: Interpreting a finitary pi-calculus in differential interaction nets. In Caires, L., Vasconcelos, V.T., eds.: CONCUR. Volume 4703 of Lecture Notes in Computer Science., Springer (2007) 333–348
5. Lafont, Y.: Interaction nets. In: POPL '90: Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, New York, NY, USA, ACM (1990) 95–108
6. Tranquilli, P.: Intuitionistic differential nets and lambda calculus. Conditionally accepted to *Theor. Comput. Sci.* (2008)
7. Regnier, L.: Lambda-Calcul et Réseaux. Thèse de doctorat, Université Paris VII (1992)
8. Di Cosmo, R., Guerrini, S.: Strong normalization of proof nets modulo structural congruences. Lecture Notes in Comput. Sci. **1631** (1999) 75–89
9. Di Cosmo, R., Kesner, D., Polonovski, E.: Proof nets and explicit substitutions. Math. Structures Comput. Sci. **13**(3) (jun 2003) 409–450
10. Pagani, M., Tortora de Falco, L.: Strong normalization property for second order linear logic. To appear on *Theor. Comput. Sci.* (2008)
11. Terese: Term Rewriting Systems. Volume 55 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (2003)
12. Danos, V., Regnier, L.: The structure of multiplicatives. Archive for Mathematical Logic **28** (1989) 181–203
13. Vaux, L.: λ-calcul différentiel et logique classique : interactions calculatoires. Thèse de doctorat, Université de la Méditerranée (2007)
14. Di Cosmo, R., Kesner, D.: Strong normalization of explicit substitutions via cut elimination in proof nets. In: LICS, IEEE Computer Society (1997) 35
15. Danos, V.: La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ-calcul). Thèse de doctorat, Université Paris VII (1990)

[12] Notice that we cannot infer CR$\sim$ of $\twoheadrightarrow$ directly from the same property in DiLL$^\circ$, as chained parallel reductions are not necessarily in DiLL$^\circ$

Technical Appendix

Let us denote by $\lambda \preccurlyeq \mu$ the relation "λ can replace μ". In the following we will use measures on free ports: these must be intended as the measures (possibly variables) of the terminal wires above such ports. Also, just for the technical appendix, we here use the notation



Lemma 6. *Given a* DiLL$^\circ$ *proof net* π, $?(e)$, $!(e)$ *and* $\mathrm{sp}(\lambda)$ *are defined and unique for all* $e \in \mathsf{w}_!(\pi)$ *and all* λ *simple subnets of* π *at any depth.* ←back

Proof. Uniqueness is clear. We prove that we can define all of the measures by a nested induction. Suppose we have defined them for all DiLL$^\circ$ proof nets of depth strictly less than $\mathrm{d}(\pi)$. Acyclicity ensures no looping dependencies can be established.

First we show that $!(e)$ is defined for all $\mathsf{w}_!(\pi)$, and at the same time that $?(e)$ and $\mathrm{sp}(\lambda)$ are for $e \in \mathsf{w}_k(\pi)$ and $\lambda \in \wp_k(\pi)$ (simple subnets at depth k) with $k \geq 1$. This is done by induction on the length of maximal ?-paths: starting with e for $e \in \mathsf{w}_0(\pi)$, or with the wire f on the principal port of the box $B \in \mathsf{b}_0(\pi)$ containing e or λ otherwise. In fact all measures inside boxes are already defined by induction hypothesis, but they depend on variables $?(p)$ and $!(p)$ with $p \in \mathsf{fp}_1(\pi)$. However by clash-freeness:

- all the variables $?(p)$ upon which the measures depend must be on the ports over a principal port of a box, and we can instantiate them with 1;
- all the variables $!(p)$ are over auxiliary ports, and we can instantiate them with $!(f)$ with $f \in \mathsf{w}_0(\pi)$, defined by secondary induction hypothesis, as these are wires further down ?-paths than the wire on the principal port.

In the other case, $e \in \mathsf{w}_0(\pi)$, $!(e)$ is defined from the !-weight on wires further down ?-paths, and possibly the spread of nets inside a box which have just been seen to be defined, all by secondary induction hypothesis.

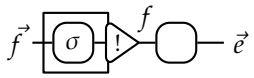
Next, $?(e)$ for $e \in \mathsf{w}_0(\pi)$ can now be defined by induction on the maximal length of !-paths starting with e, as they depend: on !-weights; on weights $?(f)$ with either $f \in \mathsf{w}_1(\pi)$ or further down !-paths; and on $\mathrm{sp}(\lambda)$ with $\lambda \in \wp_1(\pi)$.

Finally, taken any $\lambda \in \wp_0(\pi)$, its spread depends on ! and ?-weights. □

Lemma 7 (replacement). *Suppose* λ *can replace* μ, $\omega[\ ]$ *is a linear context with* $\omega[\lambda]$ *and* $\omega[\mu]$ *proof nets. Then for each* e *wire in the context* ω, $?^{\omega[\mu]}(e) \leq ?^{\omega[\lambda]}(e)$ *and* $!^{\omega[\mu]}(e) \leq !^{\omega[\lambda]}(e)$. ←back

Proof. We reason by induction on the size of ω: if there is a cell c with a wire on the internal interface, detach it from ω obtaining a smaller context ω'. One must then check that glueing c to both simple nets obtaining λ' and μ' yields

still $\mu' \preccurlyeq \lambda'$. This really poses no problem because we are adding cells with a single wire attached to the hole for now, and all the measures are defined with monotonous operations. For example in



when we check the replacement condition on $\vec{e}$ we get (if e_i is a terminal !-wire) that

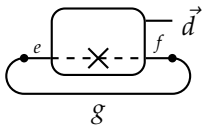
$$!^{\mu'}(e_i) = !^{\mu}(e_i)\Big[(1 + \sum !(\vec{f}))\sum_{\nu\in\sigma} \mathrm{sp}(\nu)\Big/?(f)\Big]$$

(where $[x/y]$ denotes substitution as usual) and similarly for ?, which by the hypothesis $\mu \preccurlyeq \lambda$ is less than $!^{\lambda'}$. Then checking the ?-weight for $\vec{f}$, we have

$$?^{\mu'}(f_i) = \Big((1 + \sum !(\vec{f}))\sum_{\nu\in\sigma} \mathrm{sp}(\nu)?(f^{\nu})\Big)?^{\mu'}(f)$$

which again is right as $?^{\mu'}(f) = ?^{\mu}(f) \leq ?^{\lambda}(f) = ?^{\lambda'}$.

The slightly harder part comes if ω has no cells on H_ω, and we suppose that there is a wire e which connects H_ω to itself. Supposing that glueing e to λ/μ yields an exponential wire (otherwise it is trivial), the situation can be depicted by the following picture.



As e is not a clash, e and f cannot be terminal wires of the same kind (in the sense of !-wires and ?-wires). Wlog we can suppose e a ?-wire and f an !-one (if one of the two is not exponential it is trivial). There cannot be then any exponential path linking e and f in neither λ nor μ, as it would form an exponential loop with e. Therefore the variable $!(e)$ (resp. $?(f)$) does not appear in $!^{\lambda}(f)$ and $!^{\mu}(f)$ (resp. $?^{\lambda}(e)$ and $?^{\mu}(e)$). We can therefore safely write

$$!^{\mu'}(\vec{d}) = !^{\mu}(\vec{d})\Big[!^{\mu}(p)/!(q)\Big] \leq !^{\lambda}(\vec{d})\Big[!^{\lambda}(p)/!(q)\Big] = !^{\lambda'}(\vec{d}),$$
$$?^{\mu'}(\vec{d}) = ?^{\mu}(\vec{d})\Big[!^{\mu}(p)/!(q), ?^{\mu}(q)/?(p)\Big] \leq ?^{\lambda}(\vec{d})\Big[!^{\lambda}(p)/!(q), ?^{\lambda}(q)/?(p)\Big] = ?^{\lambda'}(\vec{d}).$$

If in ω there are no cells nor wires on the hole, it means that $\omega[\lambda]$ is just a juxtaposition of λ and the net ω without the wiring on the hole. In any case it is done. □

Lemma 8 (modularity). *Take λ and $\mu_1, \ldots, \mu_n$ with simple modules and $\omega[\ ]$ a context such that $\omega[\lambda]$ and $\omega[\mu_i]$ are all* DiLL° *pure proof nets. Suppose the following points are satisfied:* ←*back*

– *for every i we have that μ_i can replace λ;*

- $\sum_i \mathrm{sp}(\mu_i) \leq \mathrm{sp}(\lambda)$;
- *if* $n = 0$, *then* $\|\lambda\| > [\,]$; *otherwise, for every* i *we have* $\left\|\mu_i\right\| + \|D_i\|^{\mu_i} < \|\lambda\|$, *where* D_i *is the set of active wires in* μ_i *that are not dormant in* λ.

Then we have the pointwise inequality

$$\left\|\omega\left[\textstyle\sum_i \mu_i\right]\right\| < \|\omega[\lambda]\|\,.$$

Proof. Let us reason by induction on the depth of the hole in $\omega[\,]$. Let $\mu = \sum_i \mu_i$.

Base step. If $\omega[\,] = \delta[\,] + \pi$ with $\delta[\,]$ linear, then it suffices to show the thesis for $\delta[\,]$. If $n = 0$ then trivially

$$\|\delta[\lambda]\| \geq \|\lambda\| > [\,] = \|0\| = \|\delta[0]\|\,.$$

Suppose therefore $n \geq 1$. By the replacement lemma we have that for every i, all the weights of wires in δ are less in $\delta[\mu_i]$ than in $\delta[\lambda]$, so $\|\delta\|^{\delta[\mu_i]} \leq \|\delta\|^{\delta[\lambda]}$. If moreover C denotes the set of cuts of $\delta[\lambda]$ that are on the interface, and C_i the same for $\delta[\mu_i]$, then clearly $C_i \subseteq C \cup D_i$, and $|C_i \setminus D_i|^{\delta[\mu_i]} \leq |C|^{\delta[\lambda]}$, as such wires are in δ and the replacement lemma applies. For the same reason when we instantiate the variables we get $\left\|\mu_i\right\|^{\delta[\mu_i]} + |D_i|^{\delta[\mu_i]} < \|\lambda\|^{\delta[\lambda]}$. We thus have

$$\begin{aligned}\left\|\omega[\mu_i]\right\| &= \|\omega\|^{\delta[\mu_i]} + |C_i|^{\delta[\mu_i]} + \left\|\mu_i\right\| \\ &\leq \|\omega\|^{\delta[\lambda]} + |C_i \setminus D_i|^{\delta[\mu_i]} + |D_i|^{\delta[\mu_i]} + \left\|\mu_i\right\| \\ &< \|\omega\|^{\delta[\lambda]} + |C|^{\delta[\lambda]} + \|\lambda\|^{\delta[\lambda]} = \|\delta[\lambda]\|\,.\end{aligned}$$

As for the spread, we have

$$\sum_i \mathrm{sp}(\delta[\mu_i]) = \mathrm{sp}^{\delta[\mu_i]}(\delta) \sum_i \mathrm{sp}^{\delta[\mu_i]}(\mu_i) \leq \mathrm{sp}^{\delta[\lambda]}(\delta) \cdot \mathrm{sp}^{\delta[\lambda]}(\lambda) = \mathrm{sp}(\delta[\lambda]).$$

So we conclude this step by

$$\left\|\delta\left[\textstyle\sum_i \mu_i\right]\right\| = \textstyle\sum_i \mathrm{sp}(\delta[\mu_i])\left\|\delta[\mu_i]\right\| < \left(\sum_i \mathrm{sp}(\delta[\mu_i])\right)\cdot\|\delta[\lambda]\| \leq \mathrm{sp}(\delta[\lambda])\,\|\lambda\| = \|\lambda\|\,.$$

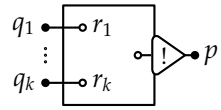
Clearly replacing $\leq$ for $<$ is still valid.

Inductive step. Suppose now that ω has the hole at non-zero depth. Let $B[\,] \in \mathsf{b}_!(\omega)$ be the box containing the hole at maximal depth, so that its contents is $\delta[\,] + \pi$ with $\delta[\,]$ linear, and let $\psi[\,]$ be the context in ω such that $\omega[\,] = \psi[B[\,]]$. If $n = 0$ by inspecting the laws of the measures one easily sees that $B[\pi] \preccurlyeq B[\delta[\lambda] + \pi]$, $\mathrm{sp}(B[\pi]) = 1 = \mathrm{sp}(B[\delta[\lambda] + \pi])$ and

$$\|B[\pi]\| = \#(B)\,\|\pi\| < \#(B)(\|\delta[\lambda]\| + \|\pi\|) = \|B[\delta[\lambda] + \pi]\|\,,$$

so we can apply inductive hypothesis and get the result.

If $n \geq 1$, applying the replacement lemma we get that the measures in $\delta[\mu_i]$ are pointwise lower than the ones in $\delta[\lambda]$, and we can instantiate them with the variables of $B[\ ]$ (and 1 for the one above the principal port). The measures in π are clearly oblivious of the changes. Let $\varrho = B[\sum_i \delta[\mu_i] + \pi]$, $\sigma = B[\delta[\lambda] + \pi]$, and

$B[\ ] =$ 

We have

$$\begin{aligned}
!^{\varrho}(p) &= (1 + \textstyle\sum_j !(q_j))\Big(\sum_i \mathrm{sp}^{\varrho}(\delta[\mu_i]) + \sum_{\nu\in\pi} \mathrm{sp}(\nu)\Big) \\
&= (1 + \textstyle\sum_j !(q_j))\Big(\mathrm{sp}^{\varrho}(\delta) \sum_i \mathrm{sp}^{\varrho}(\mu_i) + \sum_{\nu\in\pi} \mathrm{sp}(\nu)\Big) \\
&\leq (1 + \textstyle\sum_j !(q_j))\Big(\mathrm{sp}^{\sigma}(\delta) \sum_i \mathrm{sp}^{\sigma}(\lambda) + \sum_{\nu\in\pi} \mathrm{sp}(\nu)\Big) = !^{\sigma}(p), \\
?^{\varrho}(q_h) &= ?(p)(1 + \textstyle\sum_j !(q_j))\Big(\sum_i \mathrm{sp}^{\varrho}(\delta[\mu_i])?^{\varrho}(r_h) + \sum_{\nu\in\pi} \mathrm{sp}(\nu)\Big) \\
&\leq ?(p)(1 + \textstyle\sum_j !(q_j))\Big(\mathrm{sp}^{\sigma}(\delta)\Big(\sum_i \mathrm{sp}^{\varrho}(\mu_i)\Big)?^{\sigma}(r_h) + \sum_{\nu\in\pi} \mathrm{sp}(\nu)\Big) \\
&\leq (1 + \textstyle\sum_j !(q_j))\Big(\mathrm{sp}^{\sigma}(\delta)\, \mathrm{sp}^{\sigma}(\lambda)?^{\sigma}(r_h) + \sum_{\nu\in\pi} \mathrm{sp}(\nu)\Big) = ?^{\sigma}(q_h),
\end{aligned}$$

so that $\varrho \preccurlyeq \sigma$. We can see here how the sum weighted with the spread value makes sure that the measures do not grow if the number of addends inside the box grows. As for the rest of the hypotheses, $\mathrm{sp}(\varrho) = 1 = \mathrm{sp}(\sigma)$, there are no new dormant wires and in fact

$$\big\|\varrho\big\| = \#(B)\Big\|\textstyle\sum_i \delta[\mu_i] + \pi\Big\| < \#(B)\,\|\delta[\lambda] + \pi\| = \|\sigma\|\,,$$

using the base step. The box count depends solely on the variables outside the box. We can now apply inductive hypothesis, and get that

$$\Big\|\omega\Big[\textstyle\sum_i \mu_i\Big]\Big\| = \big\|\psi[\varrho]\big\| < \big\|\psi[\sigma]\big\| = \|\omega[\lambda]\|\,.$$

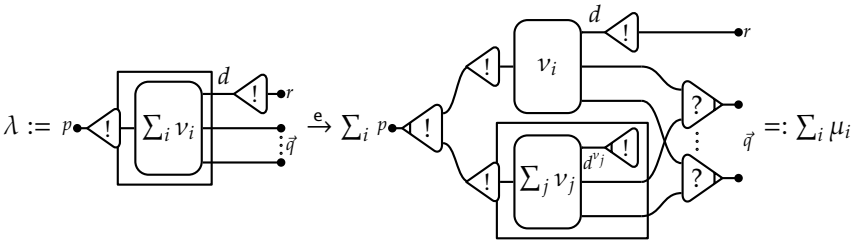
Again replacing $<$ with $\leq$ poses no problems. □

Lemma 9. $\|\,.\,\|$ *has the following properties.*

- *if* $\pi \xrightarrow{\mathtt{e}} \pi'$ *then* $\|\pi'\| < \|\pi\|$;
- *if* $\pi \xrightarrow{\mathtt{mc}} \pi'$ *then* $\|\pi'\| \leq \|\pi\|$;
- *if* $\pi \sim \pi'$ *then* $\|\pi'\| = \|\pi\|$. ←*back*

Proof. We here report some of the cases. The full list can be consulted in http://www.pps.jussieu.fr/~ptranqui/content/docs/phd.pdf.

Codereliction against box.



No new dormant wires. We may suppose $\sum_i \nu_i \neq 0$, as otherwise it is trivial. In μ_i both the ν_i outside the box and the ν_js inside get the same measures. If B is the box, then

$$\#^\lambda(B) = ?(p)(2 + \textstyle\sum_h !(q_h)) = ?(p) + \#^\mu(B) \geq 1 + \#^\mu(B).$$

Replacement hypothesis:

$$!^{\mu_i}(p) = 1 + (1 + \textstyle\sum_h !(q_h)) \sum_j \mathrm{sp}(\nu_j) \leq (2 + \textstyle\sum_h !(q_h)) \sum_j \mathrm{sp}(\nu_j) = !^\lambda(p),$$

$$\begin{aligned} ?^{\mu_i}(q_h) &= ?^{\mu_i}(q_h^{\nu_i}) + \#^{\mu_i}(B) \textstyle\sum_j \mathrm{sp}(\nu_j) ?^\mu(q_h^{\nu_j}) \\ &\leq \textstyle\sum_j \mathrm{sp}(\nu_j) ?^\lambda(q_h^{\nu_j}) + \#^{\mu_i}(B) \sum_j \mathrm{sp}(\nu_j) ?^\lambda(q_h^{\nu_j}) \\ &\leq \#^\lambda(B) \textstyle\sum_j \mathrm{sp}(\nu_j) ?^\lambda(q_h^{\nu_j}) = ?^\lambda(q_h), \end{aligned}$$

$$?^\mu(r) = 1 = ?^\lambda(r).$$

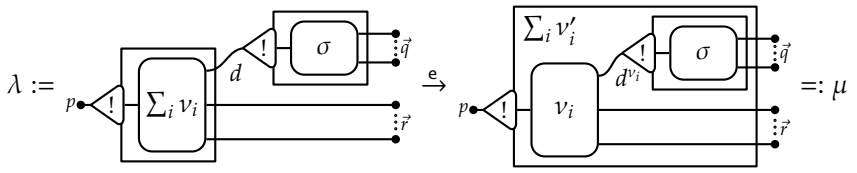
Spread hypothesis:

$$\begin{aligned} \textstyle\sum_i \mathrm{sp}(\mu_i) &= \textstyle\sum_i ?(p)\, \mathrm{sp}(\nu_i) ?^{\mu_i}(d) = ?(p) \sum_i \mathrm{sp}(\nu_i) ?^\lambda(d^{\nu_i}) \\ &\leq ?(p)(1 + \textstyle\sum_h !(q_h)) \sum_i \mathrm{sp}(\nu_i) ?^\lambda(d^{\nu_i}) = ?^\lambda(d) = \mathrm{sp}(\lambda). \end{aligned}$$

Weight hypothesis:

$$\begin{aligned} |d|^{\mu_i} &= ?^{\mu_i}(d) + 1 = ?^\lambda(d^{\nu_i}) + 1 < ?^\lambda(d) + 1 = |d|\,, \\ |d^{\nu_j}|^{\mu_i} &= ?^{\mu_i}(d^{\nu_j}) + 1 = ?^\lambda(d^{\nu_j}) + 1 < |d|\,, \\ \left\|\mu_i\right\| &\leq \|\nu_i\| + [\,|d|^\mu\,] + \#^\mu(B) \textstyle\sum_j \mathrm{sp}(\nu_j)([\,|d^{\nu_j}|\,] + |\nu_j|) \\ &\leq [\,|d|^\mu] + \#^\mu(B) \textstyle\sum_j \mathrm{sp}(\nu_j)[\,|d^{\nu_j}|\,] + (1 + \#^\mu(B)) \sum_j \mathrm{sp}(\nu_j)|\nu_j| \\ &< [\,|d|^\lambda\,] + \#^\lambda(B) \left\|\textstyle\sum_j \nu_j\right\| = \|\lambda\|\,. \end{aligned}$$

Box against box.



No new dormant wires. All measures inside both boxes are constant during this reduction. We suppose $\sum_i \nu_i \neq 0$, or else the step is trivial. Let ν_i' be the addend

inside the second box in μ, corresponding to ν_i. If B is the box with the ν_is inside in both nets then

$$\#^\lambda(B) = ?(p)(1 + !^\lambda(d) + \textstyle\sum_k(r_k)) = ?(p)(1 + (1 + \sum_h !(f_h))\,\mathrm{sp}(\sigma) + \sum_k !(r_k)) \\ > ?(p)(1 + \textstyle\sum_h !(f_h) + \sum_k !(r_k)) = \#^\mu(B),$$

where $\mathrm{sp}(\sigma) = 1$ if $\sigma = 0$, and the sum of its spreads otherwise.

Replacement:

$$\begin{aligned} ?^\mu(q_j) &= \#^\mu(B) \textstyle\sum_i \mathrm{sp}(\nu_i')?^\mu(q^{\nu_i'}) \\ &= \#^\mu(B) \textstyle\sum_i \Big(\mathrm{sp}(\nu_i)?^\mu(d^{\nu_i})(1 + \sum_h !(q_h)) \sum_{\kappa\in\sigma} ?^\mu(q^\kappa)\Big) \\ &< \Big(\#^\lambda(B) \textstyle\sum_i \mathrm{sp}(\nu_i)?^\lambda(d^{\nu_i})\Big)(1 + \sum_h !(q_h)) \sum_{\kappa\in\sigma} ?^\lambda(q^\kappa) \\ &= ?^\lambda(d)(1 + \textstyle\sum_h !(q_h)) \sum_{\kappa\in\sigma} ?^\lambda(q^\kappa) = ?^\lambda(q_j), \end{aligned}$$

$$?^\mu(r_j) = \#^\mu(B) \textstyle\sum_i \mathrm{sp}(\nu_i')?^\mu(r^{\nu_i'}) < \#^\lambda(B) \sum_i \mathrm{sp}(\nu_i)?^\lambda(r^{\nu_i}) = ?^\lambda(r_j).$$

Spread:

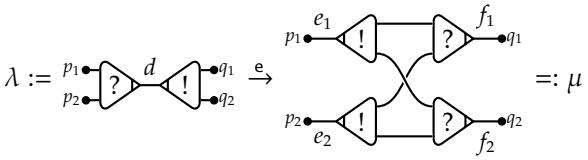
$\mathrm{sp}(\mu) = 1 = \mathrm{sp}(\lambda)$.

Weight:

$$|d^{\nu_i}|^\mu = ?^\mu(d^{\nu_i}) + !^\mu(d^{\nu_i}) < \#^\lambda(B) \textstyle\sum_i \mathrm{sp}(\mu_i)?^\lambda(d^{\nu_i}) + !^\lambda(d) = |d|^\lambda,$$

$$\begin{aligned} \|\mu\| &\le \#^\mu(B) \textstyle\sum_i \mathrm{sp}(\nu_i')\Big(\|\nu_i\| + [\,|d^{\nu_i}|^\mu\,] + ?^\mu(d^{\nu_i})(1 + \sum_h !(q_h))\,\|\sigma\|\Big) \\ &< \#^\lambda(B) \textstyle\sum_i \mathrm{sp}(\nu_i)\,[\,\|\nu_i\|\,] + \#^\lambda(B) \sum_i \mathrm{sp}(\nu_i)\,\|\nu_i\| \\ &\qquad + \Big(\#^\lambda(B) \textstyle\sum_i \mathrm{sp}(\nu_i)?^\lambda(d^{\nu_i})\Big)(1 + \sum_h !(q_h))\,\|\sigma\| \\ &< |d|^\lambda + \#^\lambda(B)\,\Big\|\textstyle\sum_i \nu_i\Big\| + ?^\lambda(d)(1 + \textstyle\sum_h !(q_h))\,\|\sigma\| = \|\lambda\|. \end{aligned}$$

Contraction against cocontraction



The new dormant wires are all e_1, e_2, f_1 and f_2.

Replacement:

$$!^\mu(p_i) = !(q_1) + !(q_2) = !^\lambda(p_i), \qquad ?^\mu(q_i) = ?(p_1) + ?(p_2) = ?^\lambda(q_i).$$

Spread:

$\mathrm{sp}(\mu) = 1 = \mathrm{sp}(\lambda)$.

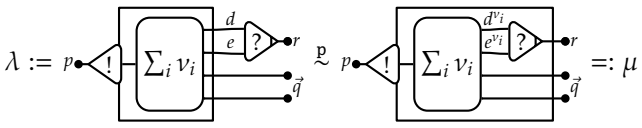
Weight:

$$|e_i|^\mu = !(p_i)(?(q_1) + ?(q_2)) < (!(p_1) + !(p_2))(?(q_1) + ?(q_2)) = |d|^\lambda,$$

$$\big|f_i\big|^\mu = ?(q_i)(!(p_1) + !(p_2)) < (?(q_1) + ?(q_2))(!(p_1) + !(p_2)) = |d|^\lambda,$$

$$\|\mu\| + \big[\,|e_1|, |e_2|, \big|f_1\big|, \big|f_2\big|\,\big] < [\,] + |d|^\lambda = \|\lambda\|.$$

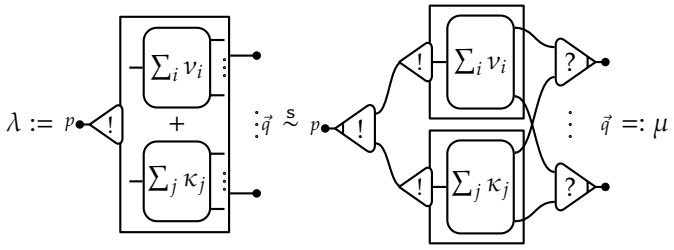
Push conversion.



Here we have only to check for $?(r)$, as the rest is trivial.

$$?^{\mu}(r) = \#^{\mu}(B)\sum_i \mathrm{sp}(\nu_i)(?^{\mu}(d^{\nu_i}) + ?^{\mu}(e^{\nu_i})) \\ = \#^{\lambda}(B)\sum_i \mathrm{sp}(\nu_i)?^{\lambda}(d^{\nu_i}) + \#^{\lambda}(B)\sum_i \mathrm{sp}(\nu_i)?^{\lambda}(e^{\nu_i}) = ?^{\lambda}(r).$$

Bang sum conversion.



The conversion does not create new dormant wires. Moreover if B is the box on the left, then $\#^{\lambda}(B)$ is the count also of both boxes on the right. Also the contents of the boxes, which we denote by B' and B'', get the same measures. Recall that such contents are required to be non zero.

Replacement:

$?^{\mu}(q_h) = \#^{\mu}(B)\Big(\sum_i \mathrm{sp}(\nu_i)?(e^{\mu}(q_h^{\nu_i})) + \sum_j \mathrm{sp}(\kappa_j)?(e^{\mu}(q_h^{\kappa_j}))\Big) = ?^{\lambda}(q_h)$,

$!^{\mu}(p) = (1 + \sum_h !(q_h))\Big(\sum_i \mathrm{sp}(\nu_i) + \sum_j \mathrm{sp}(\kappa_j)\Big) = !^{\lambda}(p)$,

Spread:

$\mathrm{sp}(\mu) = 1 = \mathrm{sp}(\lambda)$;

Weight:

$\big\|\mu\big\| = \#^{\mu}(B')\|\sum_i \nu_i\| + \#^{\mu}(B'')\|\sum_j \kappa_j\| = \#^{\lambda}(B)(\|\sum_i \nu_i\| + \|\sum_j \kappa_j\|) = \|\lambda\|$ □

Theorem 10 (finite developments). *Reduction on* DiLL$^{\circ}$ *is* SN. ←*back*

Proof. Only $\mathtt{m}$ and $\mathtt{c}$ remain to be settled. For all reductions, the pair given by $\Big(\|\pi\|, \#_{\mathtt{m}}(\pi) + \#_{\mathtt{c}}(\pi)\Big)$ strictly decreases, where $\#_{\mathtt{m}}$ just counts the multiplicative cells in π, and $\#_{\mathtt{c}}$ weights coweakenings, weakenings and boxes contining 0 in the following inductive way:

$$\#_{\mathtt{c}}(\pi) := 1 + \#_{!}(\pi) + \#_{?_0}(\pi) + \sum_{B \in \mathsf{b}_0(\pi)} (1 + \deg(B))\,\#_{\mathtt{c}}(\sigma(B))$$

where $\#_{!_0}$ and $\#_{?_0}$ count coweakenings and weakenings at depth 0, $\deg(B)$ is the number of ports of B.

– if $\pi \xrightarrow{\mathtt{e}} \pi'$ the first component strictly decreases;
– if $\pi \xrightarrow{\mathtt{m}} \pi'$ then $\|\pi'\| = \|\pi\|$, $\#_{\mathtt{m}}(\pi') < \#_{\mathtt{m}}(\pi)$ and $\#_{\mathtt{c}}(\pi) = \#_{\mathtt{c}}(\pi')$;
– if $\pi \xrightarrow{\mathtt{n}} \pi'$ then a (co)weakening disappears, so $\#_{\mathtt{c}}(\pi') < \#_{\mathtt{c}}(\pi)$, and the rest is unchanged;
– if $\pi \xrightarrow{\mathtt{p}} \pi'$ let B the box for which is happening, and B' and B'' the two corresponding boxes in π'. First note that apart the weakenings, all other cells are preserved, possibly splitted between the boxes. Let $k \geq 1$ be the number of addends in B' (i.e. the number of pulled weakenings). Then

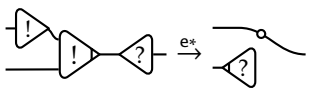
$$\begin{aligned}\deg(B)\,\#_{\mathtt{c}}(\sigma(B)) &= (1+\deg(B))\Big(k-1+\#_{\mathtt{c}}(\sigma(B'))+\#_{\mathtt{c}}(\sigma(B''))\Big)\\ &\geq (1+\deg(B))\,\#_{\mathtt{c}}(\sigma(B'))+(1+\deg(B))\,\#_{\mathtt{c}}(\sigma(B''))\\ &> (1+\deg(B'))\,\#_{\mathtt{c}}(\sigma(B'))+(1+\deg(B''))\,\#_{\mathtt{c}}(\sigma(B'')),\end{aligned}$$

where the -1 at the beginning is due to counting two times the 1 added by default to $\#_{\mathtt{c}}$. The rest is unchanged.
– if $\pi \xrightarrow{\mathtt{z}} \pi'$ then let B be the reduced box: and $\deg(B)$ (co)weakenings are created, but the deleted box weighted $1+\deg(B)$.
– if $\pi \sim \pi'$ then all multiplicative, weakening and coweakening cells are in both polynets, in boxes with the same degree, so also that also the second component is invariant. □

Proposition 11. *Reduction on* DiLL$^\circ$ *is* CR$\sim$. ←*back*

Proof. We here a couple of more cases for local confluence.

Codereliction on box on dereliction: the confluence diagram is shown in Figure 8. We use the fact that



as the other addend reduces to 0. Notice that we must use the neutral reduction of weakening.

Codereliction on box on contraction: Figure 9 shows this confluence diagram. Here we need both the neutral rule of coweakening and associativity of contraction. □

Following is a less formal description of the remaining cases (which are all accounted for).

– Box on box on dereliction, weakening or contraction, or two boxes on a third: these are in LL and therefore are known. The one with dereliction just needs to take into account the sums that may have arisen. One just takes uses the $\xrightarrow{\Sigma\mathtt{e}}$ version of the steps taken in LL.

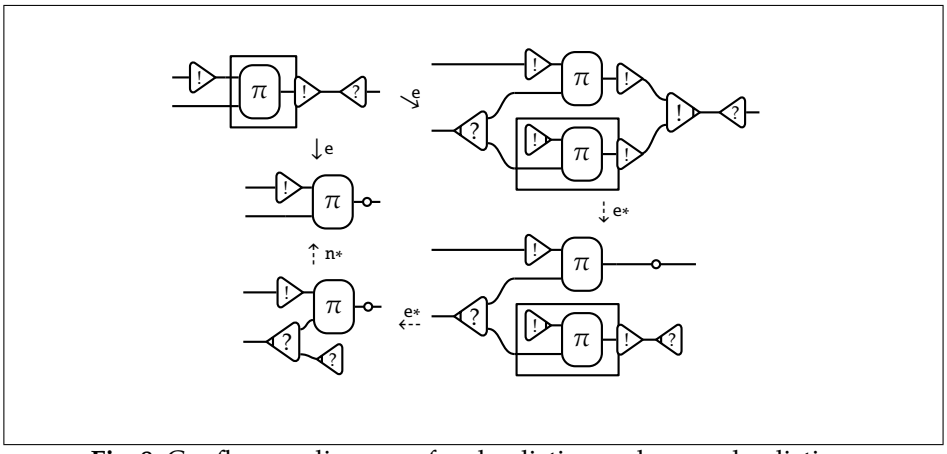


Fig. 8: Confluence diagram of codereliction on box on dereliction.

- Any combination of substituting boxes with coweakenings and cocontractions in the cases above: coweakenings and cocontractions on a box behave the same way as a box, so the confluence diagrams are identical.
- Codereliction and either box, coweakening or cocontraction on box: easy by duplicating the box, coweakening or cocontraction with the contraction created by the codereliction, and making a copy enter inside the box (all this on each addend created by the codereliction).
- Codereliction on box on weakening: both sides of the peak reduce to 0.
- Neutral reduction against a reduction on the relative (co)contraction: joinable using (co)weakening steps.
- z reducible box against (co)dereliction: bothsides reduce to 0, either by opening (a linear copy of) the 0 box or by (co)dereliction against (co)weakening.
- z reducible box on contraction or weakening: easy, the former needs n-reductions.
- z reducible box on box: straightforward, but needs a total pull reduction, as if we z-reduce the box inside we need to get the created weakenings outside.
- Every other reduction on a z-reducible box: easy, anything entering a 0 box disappears, but so does when cutting against a weakening.
- Pull reduction and a reduction on the same box but not the same wire: the diagrams are almost identical to the ones for bang sum equivalence.
- Pull reduction and a codereliction on the same wire: both reduce as if the addends in the box with the weakening were not there (they are reduced to 0).
- Pull reduction and any other cell on the same wire: on one side, the cell (box, cocontraction or coweaekening) enters the box and is erased in the addends with the weakening, possibly leaving behind multiple pull redexes. Pulling out these weakenings, together with some other c-reductions, gives the same result as the other branch.

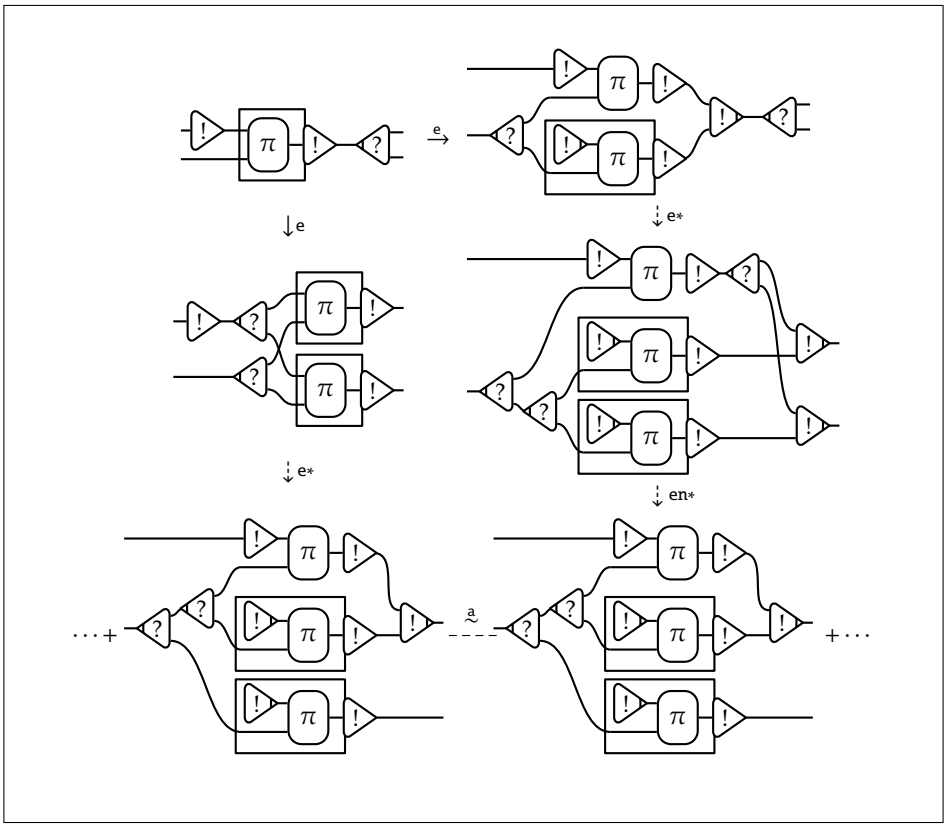


Fig. 9: Confluence diagram of codereliction on box on contraction. The + . . . part indicates a symmetric addend.

As for the coherence diagram, Figure 10 shows an example of a typical coherence diagram, serving both sides of a conversion against a certain reduction. Another list of sketched descriptions follows. The last point shows the necessity of the particular shape of neutral reduction.

- All critical pairs with associativity: trees of contractions and cocontractions are known to behave like generalized n-ary (co)contractions. In particular when dealing with a tree of two nested (co)contraction, it suffices to complete the reduction on both to join the peak. Neutrality on associativity is trivial.
- Box, coweakening or cocontraction on a s conversion: on one side they enter the box getting additively duplicated inside, while on the other they are duplicated by a contraction before entering. In any case another s conversion closes
- Dereliction on a s conversion: on one side the net is additively split by opening the box, on the other the net is split first by the dereliction on cocontraction rule, then depending on the box chosen the other gets deleted, and finally closing with n-reductions we obtain the same net.

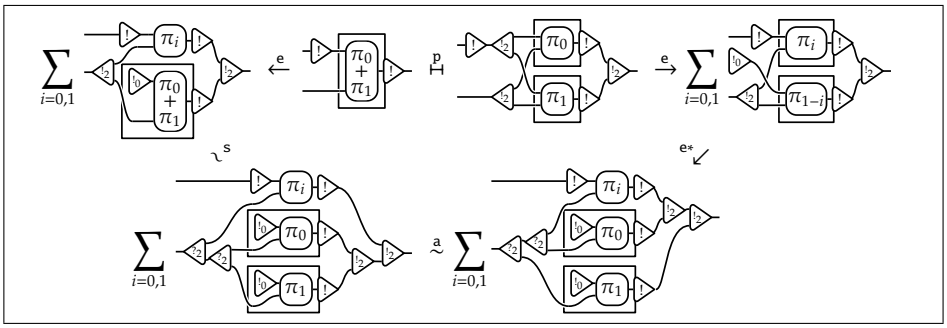


Fig. 10: Coherence diagram between the bang sum equivalence and a codereliction.

- Contraction or weakening on a s conversion: both traverse the two sides doing the same operations, on one side by being duplicated by the leading cocontraction.
- s-convertible box on another box: both sides may enter the box, safe for the trailing contractions, which must be settled by a push conversion.
- Codereliction on a s conversion: on one side, the net splits additively on the box contents on the box, with the linear copy being a single net taken from inside. On the other one, the choice of which addend of the box to extract is done at the contraction, while the other gets a coweakening exactly as with codereliction on box reduction. In any case we join in the end by s-converting the remaining boxes.
- Reduction to zero taking an addend of the conversion: on the side where the box is split, we z-reduce the box and by n-reductions we get exactly to the other side.
- Pull reduction on s-conversion: the pull is designed to integrate a sort of s-conversion step, so that it may be always performed on two sides of a conversion. n-conversion does the rest.
- Box, coweakening or cocontraction on a p conversion: the cell gets duplicated and enters the box on both sides, but in opposite order.
- Dereliction on a p conversion: the reductions on the two sides are equal.
- Contraction or weakening on a p conversion: the way in which contractions are stacked may change, but a-conversion (and n-reduction) ensures joinability.
- p-convertible box on another box: apart from having to exit/enter two boxes on one side, the two are the same.
- Codereliction on a p conversion: additive splitting and box opening happen on both sides, but in different order.
- Reduction to zero of the p-converted box: by c-normalization we get just weakenings and coweakenings on both sides.
- Pull reduction on p-conversion, on different auxiliary ports: we can p-convert all boxes involved, we just have to reorganize the stack of contractions by a-conversion.

– Pull reduction on p-conversion, with the weakening on an auxiliary port interested by the conversion: pull and neutral reductions join the two sides.
– Neutral reduction on p-conversion: n-reduction can block the conversion on one side by destroying the contraction of an addend. This is joined by a p-reduction and then by a full asp conversion. This is exaclty the point where the particular shape of the p-reduction is needed. □